APPLICATION FOR

UNITED STATES LETTERS PATENT

SPECIFICATION

INVENTOR(S):   Fumirou ABE, Masataka MATSUURA,
Masahiko NAGATA and Yasuhisa HARA

Title of the Invention: Pattern Retrieving Method, Pattern
Retrieval Apparatus,
Computer-Readable Storage Medium
Storing Pattern Retrieval Program,
Pattern Retrieval System, and
Pattern Retrieval Program

PATTERN RETRIEVING METHOD, PATTERN RETRIEVAL APPARATUS, COMPUTER-READABLE STORAGE MEDIUM STORING PATTERN RETRIEVAL PROGRAM, PATTERN RETRIEVAL SYSTEM, AND PATTERN RETRIEVAL PROGRAM

5

Background of the Invention

Field of the Invention

The present invention relates to the pattern retrieving technology for retrieving a pattern from

10 a search object, and more specifically to the pattern retrieving technology for collectively and simultaneously processing retrieval requests when the retrieval requests are received from a plurality of terminal devices, and returning

15 retrieval results to respective terminal devices, thereby exceedingly shortening the entire processing time.

Description of the Related Art

20 Conventionally, a full text retrieval system has been adopted using a character string collating method. A full text retrieval system using a character string collating method refers to a system of checking whether or not there is a

25 specified character string in the text data to be

searched while sequentially collating the specified character string with text data to be searched from the start to the end of the text data to be searched.

5   However, in the full text retrieval system using the character string collating method, the CPU of the system performs a collating operation while the CPU of the system is scanning the text data. Therefore, any other processes cannot be
10   performed during the collating operation, and it is difficult for a system, in which a plurality of user terminals are connected to a retrieval device, to provide a retrieving service to the plurality of user terminals.

15   That is, when a plurality of user terminals are connected to a retrieval device which performs a full text searching process, and each of the user terminals frequently issues a retrieval request to the device, the CPU of the retrieval device which
20   has started a retrieving process cannot perform other processes during the text scanning operation, and all other requests are kept waiting until the CPU has completed the character collating operation.

Furthermore, there has been the problem that,
25   when retrieval device receives the same retrieval

requests from different user terminals at almost the same time, it has to wastefully repeat the same retrieving process for the retrieval requests.

5 **Summary of the Invention**

The present invention has been developed to solve the above mentioned problems, and aims at providing a retrieving method, a retrieval device, a computer-readable storage medium storing a

10 retrieval program, a retrieval system, and a retrieval program capable of returning to each user terminal a retrieval result in almost the same response time as in the case where one user terminal is connected to one retrieval device

15 although retrieval requests are continuously transmitted from a plurality of user terminals to a retrieval device which is connected to the plurality of user terminals and carries out a full text search.

20 The present invention also aims at providing a retrieving method, a retrieval device, a computer-readable storage medium storing a retrieval program, a retrieval system, and a retrieval program capable of performing a retrieving process without

25 performing wasteful retrieving processes when the

same retrieval requests are received from different user terminals at almost the same time.

The pattern retrieval apparatus according to the present invention is connected to a plurality of terminal devices through a network, and includes a retrieval target data storage unit, a retrieval condition reception unit, a retrieval condition buffer unit, a retrieving process determination unit, a retrieval pattern variable table generation unit, a retrieval request expression variable table generation unit, a retrieval unit, and a transmission unit.

According to the first aspect of the present invention, the retrieval target data storage unit stores data to be searched. The retrieval condition reception unit receives the retrieval condition, transmitted from each of the plurality of terminal devices together with the terminal device information for designation of each of the terminal devices, including the retrieval pattern and the retrieval expression for retrieval of the data to be searched. The retrieval condition buffer unit stores the retrieval condition and the terminal device information received by the retrieval condition reception unit. The retrieving process

determination unit determines whether or not the preceding retrieving process is being performed. When the retrieving process determination unit determines that the preceding retrieving process is

5 not being performed, the retrieval pattern variable table generation unit generates a retrieval pattern variable table in which a retrieval pattern and a first variable having the retrieval pattern as a value are associated with each other, if there are

10 two or more identical retrieval patterns in the retrieval patterns stored in the retrieval condition buffer units, excluding the retrieval patterns other than one retrieval pattern. The retrieval request expression variable table

15 generation unit generates a retrieval request expression variable table in which the retrieval request expression indicating the retrieval pattern using the first variable and the second variable having the retrieval request expression as a value

20 are associated, and the retrieval request expression indicating the terminal device information and the retrieval expression using the second variable and the second variable having the retrieval request expression as a value are

25 associated based on the retrieval expression and

the terminal device information stored in the retrieval condition buffer unit, and the retrieval pattern variable table generated by the retrieval pattern variable table generation unit. The retrieval unit extracts a retrieval result matching the retrieval condition transmitted from each of the plurality of terminal devices by searching the retrieval target data storage unit according to the retrieval request expression variable table generated by the retrieval request expression variable table generation unit. The transmission unit transmits the retrieval result extracted by the retrieval unit to each of the plurality of terminal devices.

Brief Description of the Drawings

FIG. 1 shows the configuration of the function of the pattern retrieval system to which the present invention is applied;

FIG. 2 is a flowchart of the pattern retrieving process performed by a pattern retrieval apparatus 4 explained by referring to FIG. 1;

FIG. 3 shows a state transition table in the AC method;

FIG. 4 shows a determination table of the

determination 1 shown in FIG. 3;

FIG. 5A shows a Delta1 table used in the EBM method;

FIG. 5B shows a table for determination of the
5 right end of a pattern;

FIG. 6 shows an example of an 'erroneous read' of Japanese text;

FIG. 7 shows an example of the automaton reflecting the characteristic of code system in the
10 Japanese text;

FIG. 8 shows detecting a pattern based on the state transition;

FIG. 9 shows a concept model of the full text retrieval system in the character string collating
15 method according to the present invention;

FIG. 10 shows an example of the contents of text data to be searched;

FIG. 11 shows an example of a retrieval request;

20 FIG. 12 shows the concept of a retrieval request group;

FIG. 13 shows an example of the entire configuration of the present invention;

FIG. 14 is a flowchart showing the flow of the
25 process by the control program;

FIG. 15 is a flowchart showing the flow of the process by the reception program;

FIG. 16 is a flowchart showing the flow of the process of a requester thread;

5   FIG. 17 shows the concept of the process of writing a retrieval request to the retrieval request table;

FIG. 18 shows the concept of the process of returning a retrieval result to a terminal;

10   FIG. 19 shows the entire structure of the retrieval program;

FIG. 20 is a flowchart showing the flow of a pre-process;

FIG. 21 shows the concept of the keyword
15 variable table generating process according to the first embodiment;

FIG. 22 shows the concept of the retrieval request expression variable table 48 generating process according to the first embodiment;

20   FIG. 23 is a flowchart showing the flow of the retrieving process;

FIG. 24A shows an example (1) of a keyword variable table according to the first embodiment of the present invention;

25   FIG. 24B shows an example (1) of a retrieval

request expression variable table according to the first embodiment of the present invention;

FIG. 25A shows an example (2) of a keyword variable table according to the first embodiment of the present invention;

FIG. 25B shows an example (2) of a retrieval request expression variable table according to the first embodiment of the present invention;

FIG. 26 shows an example of a retrieval result table according to the first embodiment of the present invention;

FIG. 27 shows the contents of the text data to be searched according to the second embodiment of the present invention;

FIG. 28 shows the concept of the keyword variable table generating process according to the second embodiment of the present invention;

FIG. 29 shows the concept of generating a retrieval request expression variable table according to the second embodiment of the present invention;

FIG. 30A shows an example of the keyword variable table generating process according to the second embodiment of the present invention;

FIG. 30B shows an example of a retrieval

request expression variable table according to the second embodiment of the present invention;

FIG. 31 shows an example of a retrieval result table according to the second embodiment of the present invention; and

FIG. 32 shows loading a program into a computer according to the present invention.

Description of the Preferred Embodiments

The embodiments of the present invention are described below in detail by referring to the attached drawings.

To solve the problem, the present invention adopts the following configurations.

That is, according to an embodiment of the present invention, in a pattern retrieval system in which a plurality of terminal devices are connected to a pattern retrieval apparatus through a network, the pattern retrieving method, the pattern retrieval apparatus, the computer-readable storage medium storing a pattern retrieval program, the pattern retrieval system, and the pattern retrieval program according to the present invention receive the retrieval condition containing a retrieval pattern for retrieval of data to be searched and a

retrieval expression transmitted from each of the above mentioned plurality of terminal devices together with the terminal device information for designation of each of the terminal device, store the received retrieval condition and terminal device information in the retrieval condition buffer, determines whether or not the preceding retrieving process is being performed, generates, when it is determined that the preceding retrieving process is not being performed, a retrieval pattern variable table in which a retrieval pattern and a first variable having the retrieval pattern as a value are associated with each other, if there are two or more identical retrieval patterns in the retrieval patterns stored in the retrieval condition buffer, excluding the retrieval patterns other than one retrieval pattern, generates a retrieval request expression variable table in which the retrieval request expression indicating the retrieval pattern using the first variable and the second variable having the retrieval request expression as a value are associated, and the retrieval request expression indicating the terminal device information and the retrieval expression using the second variable and the second

variable having the retrieval request expression as a value are associated based on the retrieval expression and the terminal device information stored in the retrieval condition buffer, and the generated retrieval pattern variable table, extracts a retrieval result matching the retrieval condition transmitted from each of the plurality of terminal devices by searching the retrieval target database according to the generated retrieval request expression variable table, and transmits the extracted retrieval result to each of the plurality of terminal devices.

Thus, when there are a number of retrieval requests in a short time, the process can be performed at a much higher speed than in the conventional technology.

Furthermore, it is desired that the retrieval condition is stored in the retrieval condition buffer until it is determined that the retrieving process being performed bas been completed.

It is also desired that the retrieval condition buffer stores the retrieval condition until a predetermined time is reached or a predetermined capacity is filled.

It is also desired that the above mentioned

retrieval is performed by simultaneously retrieving a plurality of retrieval patterns.

Furthermore, it is desired that the above mentioned retrieval is performed in the Aho-
5 Corasick (AC) method, the Expanded-Boyer-Moore (EBM) method, or the Shinohara-Arikawa (SA) method.

FIG. 1 shows the configuration of the function of the pattern retrieval apparatus to which the present invention is applied.

10 In FIG. 1, a pattern retrieval system 1 comprises a plurality of terminal devices 3 connected to the pattern retrieval apparatus 4 through a electric communications circuit 2.

Each of the terminal devices 3 comprises a
15 terminal device side transmission unit 31, and a terminal device side reception unit 32.

The pattern retrieval apparatus 4 comprises a retrieval target data storage unit (database) 41, a retrieval condition reception unit 42, a retrieval
20 condition buffer unit 43, a retrieving process determination unit 44, a retrieval pattern variable table generation unit 45, a retrieval pattern variable table 46, a retrieval request expression variable table generation unit 47, a retrieval
25 request expression variable table 48, a retrieval

unit 49, and a transmission unit 50.

The terminal device side transmission unit 31 transmits the retrieval condition containing a retrieval pattern for retrieval of data to be

5   searched and a retrieval expression together with the terminal device information for designation of the terminal device 3.

The retrieval target data storage unit 41 stores the data to be searched.

10   The retrieval condition reception unit 42 receives the retrieval condition, transmitted from the terminal device side transmission unit 31 of each of the plurality of terminal devices 3 together with the terminal device information for

15   designation of each of the terminal devices 3, including the retrieval pattern and the retrieval expression for retrieval of the data to be searched.

The retrieval condition buffer unit 43 stores the retrieval condition and the terminal device

20   information received by the retrieval condition reception unit 42.   For example, the retrieval pattern is stored until the retrieving process determination unit 44 determines that the retrieving process bas been completed, or until a

25   predetermined time is reached or a predetermined

capacity is filled.

The retrieving process determination unit 44 determines whether or not the preceding retrieving process is being performed.

5 　　When the retrieving process determination unit 44 determines that the preceding retrieving process is not being performed, the retrieval pattern variable table generation unit 45 generates a retrieval pattern variable table 46 in which a 10 retrieval pattern and a first variable having the retrieval pattern as a value are associated with each other, if there are two or more identical retrieval patterns in the retrieval patterns stored in the retrieval condition buffer units 43, 15 excluding the retrieval patterns other than one retrieval pattern.

The retrieval request expression variable table generation unit 47 generates a retrieval request expression variable table 48 in which the 20 retrieval request expression indicating the retrieval pattern using the first variable and the second variable having the retrieval request expression as a value are associated, and the retrieval request expression indicating the 25 terminal device information and the retrieval

expression using the second variable and the second variable having the retrieval request expression as a value are associated based on the retrieval expression and the terminal device information

5   stored in the retrieval condition buffer unit 43, and the retrieval pattern variable table 46 generated by the retrieval pattern variable table generation unit 45.

The retrieval unit 49 extracts a retrieval

10 result matching the retrieval condition transmitted from each of the plurality of terminal devices by searching the retrieval target data storage unit according to the retrieval request expression variable table 48 generated by the retrieval

15 request expression variable table generation unit 47.   In a retrieving method, a plurality of retrieval patterns can be retrieved.   For example, the Aho-Corasick (AC) method, the Expanded-Boyer-Moore (EBM) method, or the Shinohara-Arikawa (SA)

20 method can be used.   These AC, EBM, and SA methods are briefly described later.

The transmission unit 50 transmits the retrieval result extracted by the retrieval unit 49 to each of the plurality of terminal devices 3.

25   The terminal device side reception unit 32

receives a result transmitted from the transmission unit 50.

FIG. 2 is a flowchart of the pattern retrieving process performed by the pattern
5   retrieval apparatus 4 described by referring to FIG. 1.

When the process is started, first in step S1, the retrieval condition reception unit 42 of the pattern retrieval apparatus 4 receives the
10   retrieval condition including a retrieval pattern for retrieval of data to be searched and a retrieval expression transmitted from the terminal device side transmission unit 31 of each of the plurality of terminal devices 3 together with
15   terminal device information for designation of each terminal device 3.

In step S2, the retrieval condition buffer unit 43 of the pattern retrieval apparatus 4 temporarily stores the received retrieval condition
20   and terminal device information in the buffer memory.

In step S3, the retrieving process determination unit 44 of the pattern retrieval apparatus 4 determines whether or not the preceding
25   retrieving process is being performed.

If it is being performed (YES in step S3), then control is returned to step S1, another retrieval conditions, etc. are received from the terminal device side transmission unit 31, and temporarily stored in the buffer memory (step S2). The reception (step S1) and the storage (step S2) are repeated until it is determined in step S3 that the preceding retrieving process has been completed by the retrieving process determination unit 44. Although it is determined that the preceding retrieving process has been completed, the above mentioned retrieval condition, etc. can be received/stored until a predetermined time is reached or a predetermined capacity is filled.

On the other hand, if it is determined in step S3 that a retrieving process is not being performed (that the preceding retrieving process has been completed)(NO in step S3), then, in step S4, the retrieval pattern variable table generation unit of the pattern retrieval apparatus 4 generates a retrieval pattern variable table 46 in which a retrieval pattern and a first variable having the retrieval pattern as a value are associated with each other, if there are two or more identical retrieval patterns in the retrieval patterns stored

in the retrieval condition buffer units 43, excluding the retrieval patterns other than one retrieval pattern.

Then, in step S5, the retrieval request expression variable table generation unit 47 of the pattern retrieval apparatus 4 generates a retrieval request expression variable table 48 in which the retrieval request expression indicating the retrieval pattern using the first variable and the second variable having the retrieval request expression as a value are associated, and the retrieval request expression indicating the terminal device information and the retrieval expression using the second variable and the second variable having the retrieval request expression as a value are associated based on the retrieval expression and the terminal device information stored in the buffer memory, and the generated retrieval pattern variable table 46 generated by the retrieval pattern variable table generation unit 45.

Then, in step S6, the retrieval unit 49 of the pattern retrieval apparatus 4 extracts (retrieves) a retrieval result matching the retrieval condition transmitted from each of the plurality of terminal

devices 3 by searching the retrieval target data storage unit 41 according to the generated retrieval request expression variable table 48 in the method of simultaneously retrieving a plurality

5   of retrieval patterns, that is, the AC, EBM, or SA method.

Finally, in step S7, the transmission unit 50 of the pattern retrieval apparatus 4 transmits the extracted retrieval result to each of the plurality

10   of terminal devices 3.   The terminal device side reception unit 32 of the terminal device 3 receives the above mentioned retrieval result.

Briefly described below are the above mentioned AC, EBM, and SA methods.

15   First, the AC method is briefly described. The AC method is described in detail in the reference 1 (Aho.A.V. and Corasick.M.J., 'Efficient String Matching: An Aid to Bibliographic Search', Comm.ACM, vol.18, no.6, pp.33-340, 1975).

20   The AC method is an algorithm of a pattern matching engine devised by Alfred Aho and Margaret Corasick, the authors of the above mentioned reference 1.   In this method, character strings to be searched can be detected only in one searching

25   process from start to end.

Described below is the basic logic of detecting a character string to be retrieved from character string to be searched according to the AC algorithm.

5 Normally, in the information processing technology, a character string is actually expressed by an array of binary values referred to as bits represented by 0 or 1. A character string is an array of binary numbers if it is decomposed 10 in a bit unit, but it is an array of hexadecimal numbers if it is decomposed in 4-bit units, or an array of the 256-number numeration if it is decomposed in 4-bit units.

In this example, the basic logic of the AC 15 algorithm is described with a character string processed as an array of hexadecimal numbers.

First, there are three words to be retrieved, that is, '富士 (9578 8E6D)' (two Japanese characters (their respective character codes)), '瞬 索 (8F75 20 8DF5)' (two Japanese characters (their respective character codes)), and '高 速 (8D82 91AC)' (two Japanese characters (their respective character codes)). The numbers enclosed by the parentheses are the hexadecimal representation of the words to 25 be retrieved.

Assume that the character string to be searched is '富士の瞬索は超高速 (9578 8E6D 82CC 8F75 8DF5 82CD 92B4 8D82 91AC)' (nine Japanese characters (their respective character codes)) with

5 'の (82CC)' (one Japanese character (its character code)), 'は (82CD)' (one Japanese character (its character code)), and '超 (92B4)' (one Japanese character (its character code)).

The pattern matching engine first generates a

10 state transition table as shown in FIG. 3 based on the above mentioned words to be retrieved.

In FIG. 3, it is determined in 'determination 1' whether or not an arbitrary 4 bits (the n-th four bits in the character string to be searched is

15 9 or 8, or neither 9 or 8. If it is 9, the determination 2-1 is performed. If it is 8, the determination 2-2 is performed. Otherwise, the determination 1 is performed on the (n+1)th four bits.

20 In the determination 2-1, it is determined whether or not the (n+1)th four bits refer to 5 or any of other numbers. If it is 5, the determination 3-1 is performed on the (n+2)th four bits. Otherwise, the determination 1 is performed on the

25 (n+1)th four bits.

Similarly, the determination is sequentially continued in 4-bit units.

In the above mentioned process, if the determination 8-1 is performed and the determined four bits refer to D, then the word to be retrieved '富士' (two Japanese characters (their respective character codes)) has been successfully retrieved. If the determination 8-2 is performed and the determined four bits refer to C, then the word to be retrieved '高速' (two Japanese characters (their respective character codes)) has been successfully retrieved. If the determination 8-3 is performed and the determined four bits refer to 5, then the word to be retrieved '瞬索' (two Japanese characters (their respective character codes)) has been successfully retrieved.

There are three words to be retrieved, that is, '富士' (two Japanese characters (their respective character codes)), '高速' (two Japanese characters (their respective character codes)), and '瞬索' (two Japanese characters (their respective character codes)), but the determination is not performed three times in the determination 1. Actually, the determination is performed twice whether the word refers to 9 or 8. However, since the determination

is performed whether the word refers to D or F in the determination 2-2. Therefore, although the determination frequency reduction in the determination 1 seems to be insignificant, the determination frequency can be practically reduced if the result of the determination 1 is not 8 because, in this case, the determination 2-2 can be skipped. If the probability of the occurrence of each value is equal, then the probability of the occurrence of 8 is 1/16, thereby obtaining the probability of performing the determination 2-2 is also 1/16.

In the scope of the above mentioned explanation, the character string collating time more or less depends on the number of words to be retrieved although the determination frequency is actually reduced. Especially, when the number of words to be retrieved is small, the determination frequency reduction effect is also small.

However, each determining process can be efficiently performed in the following method.

In each determining process, a determination table as shown in FIG. 4 is used. FIG. 4 shows the determination table for the determination 1. In FIG. 4, if the 4-bit value to be determined is 9, then

the following four bits are determined according to the determination table of the determination 2-1. Otherwise, the following four bits are determined according to the determination table of the determination 1.

The determination is carried out as described below according to the determination table.

Using the above mentioned character string to be searched, the first value is 9. Therefore, the column containing 9 as a 4-bit value to be determined on the determination table is directly checked, and control is passed to the determination table of the determination 2-1. That is, whatever the contents of the determination table are, all the user has to do is to check the column only. Therefore, the determining time on the determination table does not depend on the number of the words to be retrieved.

Afterwards, by scanning the character string to be searched according to the determination table, the collation between the entire character string to be searched and the entire word to be retrieved can be completed in one scanning operation.

When there is a hit on the last determination table, the collation result information such as a

hit character string, the positional information about the character string, etc. is stored in the destination table column. When control is passed to the table, the collation result information is
5   retrieved.

In the above mentioned explanation, a character is segmented in a 4-bit unit, and then a determination table is generated. If a character is segmented in a 8-bit unit, the probability of the
10 transition to the next determination table is 1/256, thereby further reducing the processing time.

Although the time required to generate a state transition table is added to a retrieving process, the time is considerably shorter than a normal
15 retrieving time. Therefore, it little affects the entire processing time.

Described briefly below is the EBM method.

The EBM method is an algorithm obtained by extending the BM (Boyer-Moore) method such that a
20 plurality of patterns can be processed. In the EMB method, like in the AC method, a plurality of patterns can be detected by scanning the text in one operation only. As compared with the AC method, the EBM method requires a smaller work area with
25 relatively high efficiency.

Described first is the outline of the BM method. In the BM method, a character string to be searched is collated with a pattern from the end to the start of the pattern. If a non-matching result

5 is output in the collating process, the collation position of the character string to be searched is shifted backward, and the collation is resumed from the end of the pattern. The process is repeatedly performed until the collation position reaches the

10 end of the character string to be searched. How many characters have to be backward skipped for collation when a non-matching result is output can be uniquely determined by the character of the character string to be searched at which the non-

15 matching result is output. Based on a given pattern, a correspondence table of a character to the number of skipped characters can be generated.

For example, when the pattern 'ABCD' is given, the table is a DELTA1 table (one-dimensional array

20 of non-zero integers) such as del_abcd shown in FIG. 5A. The upper columns refer to the characters of the character string to be searched when a non-matching result is output in the collating process, and the lower columns indicate how many characters

25 the collation position has to skip backwards. The

value of the trailing character of the pattern in the lower column is set to 0. The value of the character not existing in the pattern in the lower columns refers to the number of characters of the

5   pattern. When the collating process is performed, the trailing character of the pattern and the character at the collation position in the character string to be searched are first retrieved, and the value of the character corresponding to the

10   character in the lower column on the delta table is checked. if the value is 0, then the character matches the trailing character of the pattern. Therefore, the collating process is performed forward. If the value in the lower column is equal

15   to or larger than 1, then the trailing character of the pattern is collated with the collation positions of the character string to be searched and the pattern shifted backward by the number of characters indicated by the value. The collating

20   process is completed after the above mentioned process is repeated until the collation position reaches the end of the character string to be searched.

The table is used as follows in the EBM.

25   If two patterns 'ABCD' and 'BCDE' are given, a

deltal table (a one-dimensional array of non-negative integers) such as del_abcd and del_bcde, etc. shown in FIG. 5A is generated for each pattern. Then, these tables are combined (a smaller value in the values for the same characters is selected), and a deltal table as del_com shown in FIG. 5A is generated. When the minimum value is 0, a large value L (large: number of characters of the character string to be searched + number of characters of the pattern + 1) replaces 0. The L is used for determination of the end of the collating process.

When the trailing character of the pattern is detected (when the value of deltal is large), it is to be indicated which pattern relates to the trailing character by generating the table as shown in FIG. 5B. The figures (1, 2, 3) shown in FIG. 5B indicate respective patterns.

Using the del_com shown in FIG. 5A and 5B, the retrieving process can be completed in one collating process on a character string to be searched from the start to the end the character string although there are a plurality of patterns by performing the similar collating process of the BM method.

Finally, the SA method is briefly described.

The SA method is an algorithm of the case in which a pattern matching process is performed between a character string and Japanese text.

5 The Japanese text contains a combination of 2-byte characters and 1-byte characters. When 2-byte characters and 1-byte characters are combined, it is not possible to easily determine whether a target character code refers to a 1-byte character 10 or a 2-byte character. That is, unless the process is performed with the segmentation of a character recognized from the start of a character string, the erroneous read occurs as shown in FIG. 6.

The above mentioned AC method and the EBM 15 method disclose efficient methods of performing collating processes on a plurality of patterns, but do not disclose a method of efficiently solving the problem of the erroneous read. With Japanese text, any method cannot be effectively applied without 20 solving the problem of the erroneous read. Based on the AC method, the SA method describes means for solving the problem of the erroneous read as follows.

One of the methods for solving the problem of 25 the erroneous read is to perform a pattern matching

process with the boundary between a 1-byte character and a 2-byte character constantly recognized to prevent an erroneous read. To attain this, an automaton reflecting the characteristic of

5 a code system (shift JIS) in Japanese text is formed, and a pattern matching process is performed using it.

FIG. 7 shows an example of an automaton reflecting the characteristic of a code system in

10 Japanese text.

The automaton shown in FIG. 7 has a set of patterns {AB, 苑 (one Japanese character (its character code)), 庭 (one Japanese character (its character code))} as a target. The status 7 shown

15 in FIG. 7 indicates an intermediate status toward which the broken lines of the status 3 and 5 are led. When the status 3 and 5 cannot be changed into another status, they are first led to the intermediate status 7. After adjusting the

20 erroneous read in the intermediate status 7, the status 0 is entered. In this method, the adjustment of the erroneous read can be efficiently made in the matching process.

For example, when '外苑の初霜' (five Japanese

25 characters (their respective character codes)) is

given as text, the state transition shown in FIG. 8 occurs, and the pattern can be correctly detected.

The AC method and the EBM method are described in the reference document 2 ('Realizing five types of pattern matching methods using a function in C language, First case, English text', NIKKEI BYTE/AUGUST 1987). The SA method if described in the reference document 3 ('Realizing five types of pattern matching methods using a function in C language, Second case, Japanese text', NIKKEI BYTE/SEPTEMBER 1987).

FIG. 9 shows a concept model of the character string collating method of the full text retrieval system (hereinafter referred to as the present system).

The present system extracts the information satisfying each retrieval request from the text data to be searched in response to the retrieval requests irregularly issued from a plurality of terminals, and returns the information as a retrieval result to each terminal. FIG. 10 shows an example of the contents of the text data to be searched. The entire text data to be searched is referred to as a file. A file comprises a plurality of records, and each record comprises a plurality

of items. An item contains a character string, and a record can be identified by a record delimiter. An item in a record is segmented by an item delimiter, and can be uniquely designated by an

5   item identification code (hereinafter referred to as an item tag).

Retrieval requests are frequently received by the present system in time series. As shown in FIG. 11, each retrieval request has an array of

10 information including a combination of an item tag and a word to be retrieved, and a retrieval condition expression. These pieces of information indicate that a specified word to be retrieved exists in the item specified by the item tag in the

15 record to be searched, and that a request to search for a record satisfying a retrieval condition expression has been issued.

Then, the present system detects records satisfying a retrieval request in a file, and

20 returns a part or all of the records to a retrieval requester.

The full text retrieval system using the character string collating method as shown in FIG. 9 conventionally performs a retrieving process and

25 returns a retrieval result to a terminal each time

a retrieval request is issued by the terminal. According to the present invention, a retrieving process is performed collectively for retrieval requests from a plurality of terminals, and after

5 the retrieving process, the retrieval results are returned to the respective terminals, thereby remarkably shortening the entire processing time.

Although retrieval requests are frequently received by the present system in time series, each

10 retrieval request is not individually processed, but a plurality of retrieval requests are collectively processed. A set or retrieval requests to be collectively processed is referred to as a retrieval request group. FIG. 12 shows the concept.

15 A retrieval request group is configured as follows.

All retrieval requests received while the present system is performing the process of a retrieval request group are kept waiting, and all

20 retrieval requests waiting at the point where the process of the retrieval request group being processed is completed are defined as the next retrieval request group. If there are no waiting retrieval requests, a retrieval request group is

25 configured by the first retrieval request to be

received next. The number of retrieval requests contained in the first retrieval request group is one.

FIG. 13 shows an example of the entire configuration according to the present invention.

The present invention is configured by an information processing device, a magnetic file device, and a program storage device. The components of the program storage device are a control program, a reception program, a requester thread, a retrieval request table, a retrieval result table and a retrieval program.

FIG. 14 is a flowchart showing the flow of the process of a control program.

After performing a system initializing process on various tables, etc. (step S141), the control program is activated (step S142) and simultaneously the reception program is activated (step S143).

FIG. 15 is a flowchart showing the flow of the process of the reception program.

The reception program is constantly waiting a retrieval request from a terminal (step S151), and activates a requester thread each time a retrieval request is received from a terminal (step S152). The number of requester threads equals the number

of retrieval requests.

FIG. 16 is a flowchart showing the flow of the process of a requester thread.

After being activated, the requester thread writes a retrieval request to a retrieval request table (step S161). FIG. 17 shows the concept of the process in step S161.

Then, the requester thread enters the wait state for the completion of the retrieving process (step S162). After the retrieving process, the retrieval result is written to the retrieval result table, and the last expression variable ID indicating to which retrieval result table an answer to each requester thread has been written is written to a common area. After exiting the wait state, the requester thread refers to the last expression variable ID of its area in the common area (step S163), retrieves the contents of a hit record (step S164) from the retrieval result table, edits an answer (step S165), and returns the result to the terminal (step S166). FIG. 18 shows the concept of the processes in steps S162 through S165.

FIG. 19 shows the entire structure of a retrieval program.

A retrieval program is configured by a

preprocess, a retrieving process, a postprocess, etc.

FIG. 20 is a flowchart showing the flow of the preprocess.

The preprocess first prohibits a new requester thread from being written to the retrieval request table (step S201).

Then, based on the retrieval request table a keyword variable table and a retrieval request expression variable table are generated (steps S202 and S203), and the last expression variable ID of each requester thread is written to the common area (step S204).

Then, after deleting all contents of the retrieval request table (step S205), the prohibition of writing a requester thread from being written to the retrieval request table is released (step S206).

FIG. 21 shows the concept of a keyword variable table generating process.

A keyword variable table contains table information in which a plurality of item tags and words to be retrieved (collectively referred to as keywords) to be retrieved are arranged. A keyword is assigned a keyword variable, and has a hit flag

column indicating whether or not the keyword exists in the record to be searched. The initial value of the hit flag is 0. When there is a hit, the flag is set to 1. The hit flag is used during the
5 retrieving process.

An item tag and a word to be retrieved which are recorded in the keyword variable table are all item tags and words to be retrieved existing in the retrieval request table. However, the item tags and
10 words to be retrieved having the same contents are not recorded more than once.

FIG. 22 shows the concept of generating a retrieval request expression variable table.

A retrieval request expression variable table
15 contains table information in which a plurality of retrieval logic expressions are arranged. Each retrieval logic expression is assigned a uniquely specified code (hereinafter referred to as an expression variable). The retrieval logic
20 expression table stores every combination of a item tag and a word to be retrieved in the retrieval request table as a logical expression obtained by combining a tag variable with a word to be retrieved (practically an assigned keyword
25 variable) using a logical product (and) operator.

The expression variable assigned to the logical expression is referred to as a tag expression variable. A tag expression variable having the same contents is not entered more than once.

5   All retrieval condition expressions in the retrieval request table are replaced with a logical expression in which a tag expression variable is combined by a logical operator, and stored in the retrieval request expression variable table. The
10 expression variable assigned to the logical expression is referred to as a retrieval request expression variable. The retrieval request expression variable having a retrieval logic expression of the same contents is not entered in
15 the retrieval logic expression more than once.

The retrieval request expression variable table has a last expression variable ID column and a hit flag column. The last expression variable ID of a retrieval request expression variable is
20 assigned a number in order from 1. 0 is recorded in the last expression variable ID of a tag variable. The hit flag column is used in the retrieving process. The initial value of the hit flag column is 0.

25 After a retrieval request expression variable

table generating process, the last expression variable ID of a retrieval request expression variable corresponding to the retrieval condition of the requester thread is written to the corresponding area of each requester thread in the common area.

FIG. 23 is a flowchart showing the flow of the retrieving process.

In the retrieving process, a record satisfying a retrieval request from each requester thread is detected in the text data to be searched according to the keyword variable table and the retrieval request expression variable table, and the contents of the record are written to the retrieval result table for each last expression variable ID.

The retrieving process can be roughly divided into a pattern matching process of detecting a specified character string in a file (step S231) and a process performed based on a detected character string (steps S232 through S238).

In the pattern matching process, a record delimiter and an item delimiter are added to all keywords in the keyword variable table, and the full character string collating process is performed from the start of the text data to be

searched. In the technology of the pattern matching process according to the present invention, the technology of an AC algorithm, etc. (the above mentioned AC method, EMB method, SA method, etc.) in which the processing time is independent of the number of keywords is used.

First, when a keyword is hit (detected) in the pattern matching process, the hit flag of the keyword hit in the keyword variable table is set as true (1) (step S232).

When the item delimiter is hit, a retrieval request expression variable evaluation is performed (step S233). A retrieval request expression variable evaluation refers to a process of performing a logical operation on all expression variables in the retrieval request expression variable table, and setting the value of a hit flag as true (1).

When the second item delimiter is detected in the case where the text data to be searched shown in FIG. 10 is retrieved, the status of the keyword variable table and the retrieval request expression variable table is shown in FIGS. 24A and 24B.

After the retrieval request expression variable evaluation, all hit flags in the keyword

variable table are set as false (0) (step S234).

When a record delimiter is hit, the last expression variable ID which is true (1 or larger) which is the last expression variable ID of a

5 retrieval request expression variable whose hit flag is true (1) (YES is step S235) is retrieved, and the contents of the hit record is written into the column of the last expression variable ID of the retrieval result table (step S236). Then,

10 control is returned to step S231, and the pattern matching process is continued with all hit flags in the keyword variable table and the retrieval request expression variable table set as false (0) (step S238).

15 When the text data to be searched shown in FIG. 10 is retrieved, the status of the keyword variable table and the retrieval request expression variable table after the retrieval request expression variable evaluation after the last item delimiter

20 of the first record is detected is as shown in FIGS. 25A and 25B. After the process performed after the trailing record delimiter of the first record has been detected, the contents of the retrieval result table are as shown in FIG. 26.

25 After the retrieving process is completed, the

postprocess is performed. The postprocess releases the wait state for each requester thread.

Described above is the first embodiment of the present invention.

5   The second embodiment of the present invention is described below.

In the first embodiment of the present invention, the retrieval of data in which a record is formed by a plurality of items is described. In

10   the second embodiment, the case in which a record of data to be searched is not divided into items is described.

FIG. 27 shows the contents of the text data to be searched according to the second embodiment of

15   the present invention.

The operations of the control program, the reception program, the requester thread, and the retrieval program are the same as in the first embodiment. However, the contents of the keyword

20   variable table and the retrieval request expression variable table are different between the first and second embodiments.

When a record is not divided into items, an item tag is not used in a retrieval request.

25   Therefore, FIG. 28 shows the retrieval request

table and the keyword variable table according to the second embodiment corresponding to those according to the first embodiment shown in FIG. 21. The retrieval request expression variable table

5　according to the second embodiment corresponding to that shown in FIG. 22 has no tag variable, and is shown in FIG. 29.

Although the retrieving process is also performed as shown in FIG. 23, the contents of the

10　keyword variable table and the retrieval request expression variable table obtained after the retrieval request expression variable evaluation after detecting the last item delimiter of the first record are as shown in FIGS. 30A and 30B.

15　The retrieval result table obtained after completing the process after detecting the last record delimiter of the first record is as shown in . FIG. 31.

Described above by referring to the attached

20　drawings are the embodiments of the present invention, and the present invention has the following characteristics.

(1) When a retrieval request is issued from a terminal, a requester thread having a right to

25　communicate with the terminal is activated for the

terminal. Each requester thread writes a retrieval request from the terminal for which it has the right to communicate in one retrieval request table.

(2) When the process of a retrieval program is completed, a retrieving process is performed with the entire retrieval requests which have been written to the retrieval request table processed as one process unit.

(3) All item tags and words to be retrieved existing in the retrieval request table are collectively written into the keyword variable table regardless of a retrieval request unit. At this time, although there are the same item tags or words to be retrieved, they are not to be written more than once in the keyword variable table.

(4) Every logical product (and) condition of an item tag and a word to be retrieved, and every retrieval condition expression are collectively written into the retrieval request expression variable table. At this time, the retrieval logic expressions of the same contents are not written more than once.

(5) Numbers are sequentially written into the last expression variable ID column of the retrieval logic expression corresponding to the retrieval

condition expression in the retrieval request table. Furthermore, the assigned number is written to the common area associated with the requester thread which has written the retrieval request having the

5 retrieval condition expression, that is, the source of the retrieval logic expression.

(6) During the pattern matching process on all keywords and text data to be searched in the keyword variable table, the contents of all records

10 whose retrieval logic expression assigned to the last expression variable ID in the retrieval request expression variable table is true (1) are written into the retrieval result table corresponding to the last expression variable ID.

15 (7) The requester thread sees the last expression variable ID written to the common area, retrieves the contents of the record corresponding to the last expression variable ID from the retrieval result table, and answers the terminal

20 with which it has the right to communicate.

The embodiments of the present invention are described above by referring to the attached drawings, but the pattern retrieval apparatus to which the present invention is applied is not

25 limited to the above mentioned embodiment so far as

the functions can be realized. It can be applied to any device regardless of a single device, a system or an integrated device comprising a plurality of devices, a system for performing a process through

5   a network such as a LAN, WAN, etc.

Furthermore, the present invention can be realized by the system comprising a CPU, memory such as ROM, RAM, etc., an input device, an output device, an external storage device, a medium drive

10   device, a portable storage medium, and a network connection device connected through a bus. That is, the present invention can be attained by providing memory such as ROM, RAM, etc., external storage device, and a portable storage medium storing a

15   program code of the software for realizing the system according to the above mentioned embodiments for the pattern retrieval apparatus, and by the computer of the pattern retrieval apparatus reading the program code.

20   In this case, the program code read from the storage medium realizes a new function of the present invention, and the portable storage medium, etc. storing the program code configures the present invention.

25   As a portable storage medium for providing a

program code can be, for example, a floppy disk, a hard disk, an optical disk, CD-ROM, CD-R, DVD-ROM, DVD RAM, a magnetic tape, a non-volatile memory card, a ROM card, various storage media for storage

5 through a network connection device (that is, a communications line) such as electronic mail, personal computer communications, etc.

Furthermore, as shown in FIG. 32, the functions according to the above mentioned

10 embodiments can be realized by the functions according to the above mentioned embodiments by a computer 320 executing the program code read to memory 321, and by the OS, etc. operating in a computer performing a part or all of the actual

15 processes at an instruction of the program code.

Additionally, the functions according to the above mentioned embodiments can be realized by a program code read by a portable storage medium 322, and written to the memory 321 provided in a

20 function extension board inserted into the computer 320 and a function extension unit connected to the computer 320, and the CPU, etc. of the function extension board and the function extension unit performing a part or all of the actual processes at

25 an instruction of the program code.

That is, the present invention is not limited to the above mentioned embodiments according to the present invention, but can be realized within the scope of the gist of the present invention.

5   As described above, in the full text retrieval system using the character string collating method according to the present invention, a process can be performed at an exceedingly high speed as compared with the conventional technology when a

10   large number of retrieval request are received within a short time.

Furthermore, according to the present invention, as compared with the full text retrieval system requiring an index file, the software can be

15   considerably small, and an index file is not to be maintained, thereby realizing an operable system.

That is, when a large number of retrieval requests are received in time series within a short time in a full text retrieval system using the

20   character string collating method according to the present invention, all retrieval requests can be processed in a time shorter than the value obtained by multiplying the time required in individually processing one retrieval request by the number of

25   retrieval requests. For example, using a retrieval

engine requiring 1 second for individually processing one retrieval request, 100 retrieval requests can be completed in 3 seconds.